

Identification of functionally related enzymes by learning-to-rank methods

Michiel Stock, Thomas Fober, Eyke Hüllermeier, Serghei Glinca, Gerhard Klebe,
Tapio Pahikkala, Antti Airola, Bernard De Baets, Willem Waegeman *

Abstract

Enzyme sequences and structures are routinely used in the biological sciences as queries to search for functionally related enzymes in online databases. To this end, one usually departs from some notion of similarity, comparing two enzymes by looking for correspondences in their sequences, structures or surfaces. For a given query, the search operation results in a ranking of the enzymes in the database, from very similar to dissimilar enzymes, while information about the biological function of annotated database enzymes is ignored.

In this work we show that rankings of that kind can be substantially improved by applying kernel-based learning algorithms. This approach enables the detection of statistical dependencies between similarities of the active cleft and the biological function of annotated enzymes. This is in contrast to search-based approaches, which do not take annotated training data into account. Similarity measures based on the active cleft are known to outperform sequence-based or structure-based measures under certain conditions. We consider the Enzyme Commission (EC) classification hierarchy for obtaining annotated enzymes during the training phase. The results of a set of sizeable experiments indicate a consistent and significant improvement for a set of similarity measures that exploit information about small cavities in the surface of enzymes.

1 Introduction

Modern high-throughput technologies in molecular biology are generating more and more protein sequences and tertiary structures, of which only a small fraction can ever be experimentally annotated w.r.t. functionality. Predicting the biological function of enzymes¹ remains extremely challenging, and especially novel functions are hard to detect, despite the large number of automated annotation methods that have been introduced in the last decade [1, 2].

Existing online services, such as BLAST or ReliBase, often provide tools to search in databases that contain collections of annotated enzymes. These systems rely on some notion of similarity when searching for related enzymes, but the definition of similarity differs from system to system. Indeed, a vast number of measures for expressing similarity between two enzymes exists in the literature, performing calculations on different levels of abstraction. One can make a major subdivision of these measures into approaches that solely use the sequence of amino acids, approaches that also take into account the tertiary structure, and approaches that consider local fold information by analyzing small cavities (hypothetical binding sites) at the surface of an enzyme.

Sequence-based measures such as BLAST [3] and PSI-BLAST [4] can be computed in an efficient manner and are able to find enzymes with related functions under certain conditions. In addition to these, several kernel-based methods have been developed to make predictions for proteins at the sequence level - see e.g. [5, 6]. A high sequence similarity usually results in a high structural similarity, and proteins with a sequence identity (the number of matches in an alignment) above 40 % are generally considered to share the same structure [7]. However, this assumption becomes less reliable in the twilight zone, when the sequence identity is situated between 25 and 40 %. Furthermore, enzymes with comparable functions can exhibit sequences with very low sequence identity [8, 9].

For these reasons, and because three-dimensional crystal structures are becoming more and more available in online databases, the comparison of proteins at the structural level has gained increasing attention. The secondary

*M. Stock, B. De Baets and W. Waegeman are with the Department of Mathematical Modelling, Statistics and Bioinformatics, Ghent University, Coupure links 653, 9000 Ghent, Belgium, email: firstname.surname@ugent.be. T. Fober and E. Hüllermeier are with the Philipps-Universität of Marburg, Department of Mathematics and Computer Science, Hans-Meerwein-Straße 6, D-35032 Marburg, Germany. S. Glinca and G. Klebe are also with the Philipps-Universität of Marburg, Department of Pharmacy, Marbacher Weg 6-10, D-35032 Marburg, Germany. T. Pahikkala and A. Airola are with the Department of Information Technology and the Turku Centre for Computer Science, University of Turku, Joulukaisenkatu 3-5 B 20520 Turku, Finland.

¹Enzymes are biomolecules that catalyze chemical reactions. All the enzymes we consider in this work are proteins and vice versa. Both notions will be used interchangeably.

structure of an enzyme is known to highly influence its biological function [10] and contains valuable information that is missing at the sequence level [11, 12, 13]. Many approaches that perform calculations on the overall fold of the protein have been developed - see e.g. [14, 15, 16]. Unfortunately, such approaches are also not optimal for determining the function of enzymes. They require knowledge of active site residues and usually lead to a quite coarse representation, especially for enzymes, where often only a few specific residues are responsible for the catalytic mechanism [17]. For example, the vicinal-oxygen-chelate superfamily shows a large functional diversity while having only a limited sequence diversity [18, 19]. It has also been shown that some parts of the protein structure space have a high functional diversity [20], further limiting the use of global fold similarity. For these reasons, many methods consider local structural features, such as evolutionary conserved residues [21, 22].

The most appropriate similarity measures for the prediction of enzyme functions focus on surface regions in which ligands, co-factors and substrates can bind [23]. Cavities in the surface are known to contain valuable information, and exploiting similarities between those cavities helps finding functionally related enzymes. By considering structural and physico-chemical information of binding sites, one can detect relationships that cannot be found using traditional sequence- and fold-based methods, making such similarities of particular interest for applications in drug discovery [24, 25]. In addition to providing a complementary notion of protein families [26], these methods also allow for extracting relationships between cavities of unrelated proteins [27]. Similarity measures that highlight cavities and binding sites can be further subdivided into graph-based approaches such as [28, 29, 30, 31, 32], geometric approaches such as [33, 34] and feature-based approaches such as [35, 36]. These measures will be discussed more thoroughly in Section 2.

This paper aims to show that the search for functionally related enzymes can be substantially improved by applying *learning-to-rank* methods. These algorithms use training data to build a mathematical model for ranking objects, such as enzymes, that are not necessarily seen among the training data. While our methods can be applied to all types of data, as long as a meaningful similarity measure can be constructed, we only demonstrate its power using cavity-based measures, for the reasons explained above. Ranking-based machine learning algorithms are often used for applications in information retrieval [37]. Due to their proven added value for search engines, ranking-based machine learning methods have gained some popularity in bioinformatics, for example in drug discovery [38, 39] or to find similarities between proteins [40, 41]. Despite this, many online services such as BLAST, PDB, Dali and CavBase solely rely on similarity measures to construct rankings, without utilizing annotated enzymes and learning algorithms to steer the search process during a training phase. However, due to the presence of annotated enzymes in online databases, improvements can be made by applying ranking-based machine learning algorithms. This amounts to a transition from an unsupervised to a supervised learning scenario.

Using four different cavity-based similarity measures and one based on sequence alignment as input for RankRLS [42], a kernel-based ranking algorithm, we demonstrate a significant improvement for each of these measures. RankRLS works in a similar way as competitors such as RankSVM [43], because it uses annotated training data to learn rankings during a training phase. The training data is annotated via the Enzyme Commission (EC) functional classification hierarchy, a commonly used way to subdivide enzymes into functional classes. EC numbers adopt a four-number hierarchical structure, representing different levels of catalytic detail. Importantly, this representation focuses on the chemical reactions that are performed, and not on structure or homology. As explained more elaborately in Section 2, the EC numbers are used to construct a ground-truth catalytic similarity measure, and subsequently to generate ground-truth rankings. In addition to obtaining annotated training data, this procedure also allows for a fair comparison with the more traditional approach, using conventional performance measures for rankings. This way of evaluating also characterizes the difference between our search engine approach and previous work in which supervised learning algorithms for EC number assignment have been considered – for a far from complete list see e.g. [44, 29, 45, 46, 47]. In this work we are unable to compare to such methods, because they do not return rankings as output. Nonetheless, similar to some of these approaches, we do take the hierarchical structure of the EC numbers into account. Instead of predicting one EC number, a ranking of functionally related enzymes is returned for a given query. In this scheme the top of the obtained ranking is expected to contain enzymes with functions similar to the query enzyme with an unknown EC number. A ranking provides end users with a generally well-known and easily understandable output, while still useful results can be retrieved when an enzyme with a new EC number is encountered.

2 Material and methods

2.1 Database

Our work builds upon CavBase, a database that is made commercially available as part of ReliBase [48]. CavBase can be used for the automated detection, extraction, and storage of protein cavities from experimentally determined

protein structures, which are available through the Protein Data Bank (PDB).

The geometrical arrangement of the pocket and its physico-chemical properties are first represented by pre-defined pseudocenters – spatial points that characterize the geometric center of a functional group specified by a particular property. The type and the spatial position of the pseudocenters depend on the amino acids that border the binding pocket and expose their functional groups. They are derived from the protein structure using a set of predefined rules [49]. Hydrogen-bond donor, acceptor, mixed donor/acceptor, hydrophobic aliphatic, metal ion, pi (accounts for the ability to form π - π interactions) and aromatic properties are considered as possible types of pseudocenters. These pseudocenters can be regarded as a compressed representation of surface areas where certain protein-ligand interactions are encountered. Consequently, a set of pseudocenters is an approximate representation of a spatial distribution of physico-chemical properties.

To build and test our models we require an appropriate data set that contains sufficiently many proteins and EC classes. Based on the experience of local pharmaceutical experts, we chose the data set of EC classes depicted in Table 1. To generate the first data set (data set I), we retrieved all proteins from the PDB which got assigned one of these EC classes. Thus, we ended up with a set of 5,257 proteins. To ensure that only unique proteins were contained in our data set, we used the protein culling server² with its default parameterization. As such, all proteins that have high pairwise homology were filtered out. This procedure resulted in a data set of cardinality 1,714. To extract the active site of the protein we used the assumption that the largest binding site of a protein does contain its catalytic center [23]. Hence, for each protein we took the binding site from the database CavBase which maximized the volume. From our data set, 158 proteins were not contained in the CavBase (e.g., because the structure was determined by NMR instead of X-ray). Therefore these proteins were removed from the data set, resulting in a final data set of size 1,556.

The first data set comes with two drawbacks. First of all, the binding site containing the catalytic centre was determined by a pure heuristic, namely by taking the largest binding site among all binding sites a protein exhibits. Moreover, sufficient resolution was not a criterium for selecting the cavities. This may lead to a data set of low quality. Therefore, relying on the expertise of pharmaceutical experts we compiled another data set referred to as data set II, containing the same EC classes. For this data set, all proteins from the PDB that have a resolution of at least 2.5 Å were considered. Moreover the binding site volume was required to range between 350 Å³ and 3500 Å³. Structures not meeting these conditions were eliminated since resolutions below 2.5 Å usually lead to a too coarse representation, while binding sites with volumes outside the above-mentioned range are usually artefacts produced by the algorithm used for their detection. From the resulting set of 24,102 proteins the active site was selected. This resulted in a data set of 1730 enzymes on which we applied the protein culling server to finally end up with a second data set of 561 enzymes.

A pairwise sequence similarity matrix and phylogenetic tree of our data sets can be found in the supplementary materials.

2.2 Similarity measures for cavities

In the introduction we have motivated why our analysis is restricted to similarity measures for cavities, which are three-dimensional objects that can be represented in multiple ways. Some measures are graph-based, transforming cavities into node-labeled and edge-weighted graphs. This allows to apply traditional techniques to compare graphs (e.g. [28]). Unfortunately, techniques that construct a boolean similarity measure, such as those based on graph isomorphisms, are not appropriate for comparing noisy and flexible protein structures. Computing the maximum common subgraph [50] can be considered as a more appropriate alternative, and this method will be used in this paper as a baseline (see below). The graph edit distance [51] is another measure to compare graphs, specifying the number of edit operations needed to transform a given graph into another graph. This distance can be calculated in different ways, e.g., by using a greedy heuristic [52] or quadratic programming [53]. Unfortunately, the graph edit distance is very hard to parameterize and often quite inefficient. More efficient approaches belong to the class of graph kernels. They have gained a lot of attention in bioinformatics, as they allow for a sufficiently high degree of error tolerance. Different realizations are available, such as the shortest path kernel [54], the random walk kernel [30] and the graphlet kernel [31, 32]. Graph kernels work particularly well for small molecules such as ligands, but they are less useful for larger molecules such as proteins. They gave rather poor results in [36], which explains why we concentrated here on the maximum common subgraph as a representative for graph-based approaches.

As a second category of measures for cavities, geometric methods directly process the labeled spatial coordinates of the functional parts, denoted as point clouds, instead of transforming a protein cavity into a graph. Remarkably, only few approaches have been proposed that build on this representation. In [33] geometric hashing is employed to

²<http://www.bioinf.manchester.ac.uk/leaf/>

Table 1: List of the 21 EC numbers with their accepted name and the number of examples of each class for the two data sets.

EC number	accepted name	# set I	# set II
EC 1.1.1.1	alcohol dehydrogenase	23	15
EC 1.1.1.21	aldehyde reductase	35	30
EC 1.5.1.3	dihydrofolate reductase	110	6
EC 1.11.1.5	cytochrome-c peroxidase	92	31
EC 1.14.15.1	camphor 5-monooxygenase	30	36
EC 2.1.1.45	thymidylate synthase	63	22
EC 2.1.1.98	diphthine synthase	5	43
EC 2.4.1.1	phosphorylase	43	40
EC 2.4.2.29	tRNA-guanine transglycosylase	32	16
EC 2.7.11.1	non-specific serine/threonine enzyme kinase	304	24
EC 3.1.1.7	acetylcholinesterase	23	13
EC 3.1.3.48	enzyme-tyrosine-phosphatase	151	28
EC 3.4.21.4	trypsin	118	72
EC 3.4.21.5	thrombin	87	51
EC 3.5.2.6	β -lactamase	153	8
EC 4.1.2.13	fructose-bisphosphate aldolase	48	4
EC 4.2.1.1	carbonate dehydratase	186	76
EC 4.2.1.20	tryptophan synthase	13	7
EC 5.3.1.5	xylose isomerase	18	21
EC 5.3.3.1	steroid Δ -isomerase	14	10
EC 6.3.2.1	pantoate- β -alanine ligase	8	8

calculate a superposition of protein cavities that can be used to derive an alignment and a similarity score. A similar approach was used in [34], in which an optimization problem was solved instead of applying geometric hashing. Beside these two approaches, several other methods exist for comparing two point clouds [55]. Unfortunately, the majority of these methods cannot cope with biological data, due to a very high complexity or error intolerance.

As a third family of approaches, one can also represent the protein cavity as a feature vector, taking both the geometry of the cavity and physico-chemical properties into account – see e.g. [36, 35]. Subsequently, traditional or specialized measures can be applied on these vectors to obtain similarity scores between protein cavities [56, 57].

In the experiments we selected representative method for each of the three groups: one graph-based measure, one geometric measure and one feature-based measure. We also considered the original CavBase measure and a measure obtained from the Smith-Waterman protein sequence alignment. This lead to a comparison of five different measures, four based on cavities and one based on sequence alignment. Below, these measures are explained more in detail:

Labeled Point Cloud Superposition (LPCS) [34]. This value is obtained by processing labeled point clouds.

Hence, the CavBase data can be used directly without a need for transforming it into another representation. Intuitively, two labeled point clouds are considered similar if they can be spatially superimposed. More specifically, an approximate superposition of the two structures is obtained by fixing the first point cloud and moving the second point cloud as a whole. Two point clouds are well superimposed when each point in the first cloud can be matched with a point in the second point cloud, while the distances of these points are small and their labels consistent. This concept is used to define a fitness function that is maximized using a direct search approach [58]. The obtained maximal fitness is taken as the similarity between the two labeled point clouds. A similar measure was also proposed in [59], but a convolution kernel is suggested to obtain similarities between the point clouds.

Maximum Common Subgraph (MCS) [50]. Using the MCS, the original representation in the form of a labeled point cloud must be transformed into a node-labeled and edge-weighted graph. Each pseudocenter is becoming a node labeled with the corresponding physico-chemical property. To capture the geometry, a complete graph is considered, where each edge is weighted with the Euclidean distance between the two pseudocenters it is adjacent to. The problem of measuring similarity between protein cavities now boils down to the problem of measuring similarity between graphs. A well-known approach here is to search for the maximum common subgraph of the two input graphs and to define similarity as the size of the maximum common subgraph relative to the size of the larger graph. In case of noisy data, a threshold ϵ is required,

defining two edges as equal if their weight differs at most by ϵ . In this paper, this parameter is set to 0.2 Å, as recommended by several authors [52, 60].

CavBase (CB) similarity [49]. CavBase also makes use of an algorithm for the detection of common subgraphs. Instead of considering the largest common subgraph, as done in the case of MCS, the 100 largest common subgraphs are considered. Each common subgraph is used to determine a transformation rule by means of the Kabsch algorithm [61], which superimposes both proteins. In a post-processing step the surface points are also superimposed according to the transformation rule, and a similarity score is derived using these surface points. Eventually, a set of 100 similarity values is obtained, from which the highest value is returned as similarity between the two protein cavities.

Fingerprints (FP) Fingerprints are a well-known concept and have been used successfully in many domains. For the comparison of protein binding sites, the authors in [62] transformed the protein binding site into a node-labeled and edge-weighted graph as described above. Moreover they defined generically a set of features, namely complete node-labeled and edge-weighted graphs of size 3. For each such feature, a test is performed to decide whether or not the feature is contained in the graph representing the protein. This is done by subgraph isomorphism, to check whether the labels are identical. The nodes of the features are labeled by the set of physiochemical properties. Edges of patterns are labeled by intervals or bins and instead of testing for equivalence, a test is performed whether edge weight of the graph representing the protein falls into the bin of the pattern. The thus generated fingerprints are compared by means of the Jaccard similarity measure, as proposed by [56].

Smith-Waterman (SW) Beside using structure-based approaches to compare protein binding sites, we used also sequence alignment in our experimental study. To calculate sequence alignments we used the Smith-Waterman algorithm [63] which was parameterized with the Blosom-62 matrix. From the sequence alignment we derived the sequence identity which was subsequently used to perform experiments.

2.3 Unsupervised ranking

In the introduction we have explained why existing online services such as BLAST, PDB, Dali and CavBase construct rankings in an unsupervised way. These systems create a ranking by means of a similarity measure only, without training a model that uses annotated enzymes. The annotated enzymes in a database are simply ranked according to their similarity with an enzyme query with unknown function. In the case of CavBase, the enzymes having a high cavity-based similarity appear on top of the ranking, those exhibiting a low cavity-based similarity end up at the bottom. More formally, let us represent the similarity between a pair of enzymes by $K : \mathcal{V}^2 \rightarrow \mathbb{R}^+$, where \mathcal{V} represents the set of all potential enzymes. Given the similarities $K(v, v')$ and $K(v, v'')$, we compose the ranking of v' and v'' conditioned on the query v as:

$$v' \succeq_v^K v'' \Leftrightarrow K(v, v') \geq K(v, v''), \quad (1)$$

where \succeq_v^K indicates the relation *is ranked higher than, for query v , based on the similarity K* . Note that this is a relation between two enzymes *conditioned* on a third enzyme. In our context, there is no meaningful ranking possible between enzymes v' and v'' without referring to another enzyme v . This approach adopts the same methodology as a nearest neighbor classifier, but a ranking rather than a class label should be seen as the output of the algorithm.

The quality of such rankings can be evaluated when the database contains annotated enzymes and annotated queries. In an evaluation phase, we compare the obtained ranking with the *ground truth* ranking, which can be constructed from the EC numbers for annotated enzymes. This ground truth ranking can be deduced from the catalytic similarity (i.e., ground truth similarity) between the query and all database enzymes, by counting the number of successive matches in the EC label of the query and the database enzymes. Thus the catalytic similarity is a property of only a pair of enzymes. In contrast, in order to create the ground truth ranking of two enzymes, the catalytic similarity has to be calculated w.r.t. a third enzyme. For example, an enzyme with EC number EC 2.4.2.23 has a catalytic similarity of two compared to an enzyme labeled as EC 2.4.99.12, since both enzymes belong to the family of glycosyltransferases. Conversely, the same enzyme manifests a similarity value of only one with an enzyme labeled as EC 2.8.2.23. Both are transferases in this case, but they show no further relevant similarity in the chemistry of the reactions to be catalyzed.

More formally, let us represent the catalytic similarity between two enzymes by a relation $Q : \mathcal{V}^2 \rightarrow \{0, 1, 2, 3, 4\}$. Q is defined by:

$$Q(v, v') = \sum_{i=1}^4 \prod_{j=1}^i q_i q_j,$$

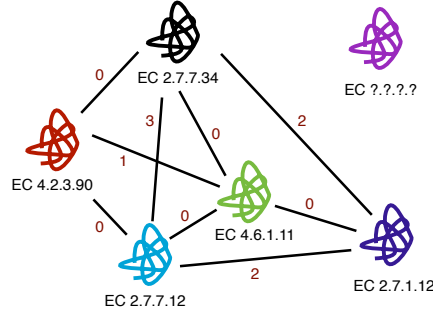


Figure 1: Six enzyme structures are shown, five of which correspond to a known EC number. The catalytic similarity Q is depicted on the edges of the graph. The algorithm that we present allows us to infer for the unannotated query (denoted as EC ??.?.?) a ranking of the annotated enzymes. To this end, the unsupervised approach solely uses cavity-based similarity measures, whereas the supervised approach also takes the EC numbers of annotated enzymes into account.

where q_i equals 1 if the i th digit of the EC numbers of v and v' are the same and 0 otherwise. Figure 1 gives an example for six enzymes, five of which correspond to a known EC number. The catalytic similarity Q is depicted on the edges of the graph. The proposed algorithm allows us to infer for an unannotated query a ranking of the annotated enzymes, some of which the algorithm may not have encountered among the training data.

Given the similarities $Q(v, v')$ and $Q(v, v'')$, we compose similar to Eq. (1) the ground truth ranking of v' and v'' conditioned on the query v as:

$$v' \succeq_v^Q v'' \Leftrightarrow Q(v, v') \geq Q(v, v'').$$

As a result, an entire ground truth ranking of database enzymes with known EC numbers can be constructed, given an annotated query enzyme.

2.4 Supervised ranking

In contrast to unsupervised ranking approaches, supervised algorithms do take ground truth information into account during a training phase. We perform experiments with so-called conditional ranking algorithms [64, 65] using the RankRLS implementation [42]. Let us introduce the short-hand notation $e = (v, v')$ to denote a couple consisting of an enzyme query v and a database enzyme v' . RankRLS produces a linear basis function model of the type:

$$h(e) = h(v, v') = \langle \mathbf{w}, \Phi(e) \rangle, \quad (2)$$

in which \mathbf{w} denotes a vector of parameters and $\Phi(e)$ an implicit feature representation for the couple $e = (v, v')$.

RankRLS differs from more conventional kernel-based methods because it optimizes a convex and differentiable approximation of the rank loss in bipartite ranking (i.e., area under the ROC curve) instead of the zero-one loss. Together with the standard L2 regularization term on the parameter vector \mathbf{w} and a regularization parameter λ , the following loss is minimized:

$$\mathcal{L}(h, T) = \sum_{v \in V} \sum_{e, \bar{e} \in E_v} (Q_e - Q_{\bar{e}} - h(e) + h(\bar{e}))^2, \quad (3)$$

for a given training set $T = \{(e, Q_e) \mid e \in E\}$. Here $Q_e = Q(v, v')$ denotes the ground truth similarity as defined above, E the set of training query-object couples for which ground truth information is available and E_v the subset of E containing results for the query v . The outer sum in Eq. (3) takes all queries into account, and the inner sum analyzes all pairwise differences between the ranked results for a given query. This loss can be minimized in a computationally efficient manner, using analytic shortcuts and gradient-based methods, as shown in [64, 65].

According to the representer theorem [66], one can rewrite Eq. (2) in the following dual form:

$$h(e) = \langle \mathbf{w}, \Phi(e) \rangle = \sum_{\bar{e} \in E} a_{\bar{e}} K^{\Phi}(e, \bar{e}).$$

with $K^{\Phi}(e, \bar{e})$ a kernel function with four enzymes as input and $a_{\bar{e}}$ the weights in the dual space. In this paper we adopt the Kronecker product feature mapping, containing information on couples of enzymes:

$$\Phi(e) = \Phi(v, v') = \phi(v) \otimes \phi(v'),$$

with $\phi(v)$ a feature mapping of an individual enzyme and \otimes the Kronecker product. One can easily show that this pairwise feature mapping yields the Kronecker product pairwise kernel in the dual representation:

$$K^\Phi(e, \bar{e}) = K^\Phi(v, v', \bar{v}, \bar{v}') = K^\phi(v, \bar{v})K^\phi(v', \bar{v}'),$$

with K^ϕ a traditional kernel for enzymes. Specifying a universal kernel for K^ϕ leads to a universal kernel for K^Φ [67], indicating that one can use the kernel to represent any arbitrary relation, provided that the learning algorithm has access to training data of sufficient quality. This kernel has been introduced in [68] for modelling protein-protein interactions. We consider this kernel because of its universal approximation property, but also other pairwise kernels exist, such as the cartesian pairwise kernel [69], the metric learning pairwise kernel [70] and the transitive pairwise kernel [71, 64]. Nonetheless, it is probably not very surprising that such kernels only yield an improvement if the concepts to be learned satisfy the restrictions that are imposed by the kernels [67].

With the exception of the FP measure, none of the similarity measures discussed in Section 2.2 are strictly speaking valid kernels. Using the above construction, all similarity measures can be converted into kernels of type K^ϕ , when they are made symmetric and positive definite. These attributes guarantee a numerically stable and unique solution of the learning algorithm. We simply enforced symmetry by averaging the similarity matrix with its transpose. Subsequently, we made the different similarity matrices positive definite by performing an eigenvalue decomposition and setting all eigenvalues smaller than 10^{-10} equal to zero. This method leads to a negligible loss of information compared to the numerical accuracy of our algorithms and data storage. Finally, each kernel matrix was normalized so that all diagonal elements have a value equal to one. Since these procedures were performed on the whole data set, one arrives at a so-called transductive learning setting [72]. Minor adjustments would obtain a more traditional inductive learning setting. Note that overfitting is prevented when applying this procedure, since the EC numbers of the enzymes in the data set are not taken into account.

Since the catalytic similarity is a symmetric measure we also perform a post-processing to the output of our algorithm. The matrix with the predicted values used for ranking the enzymes is made symmetric by averaging it with its transpose.

2.5 Performance measures for ranking

The ranking obtained with unsupervised or supervised learning algorithms can be compared to the ground truth ranking by applying performance measures that are commonly used in information retrieval.

First of all, the ranking accuracy (RA) is considered, and it is defined as follows:

$$RA = \frac{1}{|V|} \sum_{v \in V} \frac{1}{|\{E_v \mid y_e > y_{\bar{e}}\}|} \sum_{\substack{e, \bar{e} \in E_v: \\ y_e < y_{\bar{e}}}} I(h(e) - h(\bar{e})),$$

with $I(x)$ the Heaviside function, returning one when its argument is positive, zero when its argument is negative and 1/2 when its argument is zero. The ranking accuracy can be considered as a generalization of the area under the ROC curve for more than two ordered classes [73].

Our interest in the ranking accuracy is motivated by two reasons. Firstly, unlike most other performance measures we consider, all levels of the EC hierarchy are taken into account to determine the performance of different algorithms. Predicted rankings can be interpreted as layered or multipartite rankings – see e.g. cite-Waegeman2008e,Furnkranz2009. The ranking accuracy preserves this hierarchical structure by counting all pairwise comparisons. The second reason of interest is based on the fact that the ranking accuracy is optimized by the RankRLS software, using the convex and differentiable approximation given in Eq. (3). This loss function characterizes the most important difference with more traditional kernel-based algorithms, such as support vector machines, resulting in an information retrieval setting instead of a more traditional classification or network inference setting.

Since the ranking accuracy is not generally known in bioinformatics, we also evaluated our algorithms using three more conventional performance measures that are commonly considered for bipartite rankings (i.e., rankings containing relevant versus irrelevant objects). These three measures are the area under the ROC curve (AUC), mean average precision (MAP) and normalized discounted cumulative gain (nDCG) [74]. For AUC and MAP all ground truth rankings had to be converted into bipartite rankings, leading to a decrease in granularity for performance estimation. We chose a cut-off threshold of three in ground truth similarity: a retrieved enzyme is relevant to the enzyme query if at least the first three parts of its EC number are identical to the query.

2.6 Experimental setup

We selected two data sets of enzymes from CavBase, as described in Section 2.1. The ground-truth catalytic similarity of all enzyme pairs was computed for each data set. Each data set was further randomized and split in

Table 2: Summary of the results obtained for unsupervised and supervised ranking for both data sets. For each combination of similarity and type of performance measure, the performance is averaged over the different folds and queries, with the standard deviation between parentheses.

		Set I				
		CB	FP	MCS	LPCS	SW
Unsupervised	RA	0.6096 (0.1286)	0.6411 (0.1670)	0.5960 (0.1010)	0.6173 (0.1244)	0.6576 (0.1446)
	MAP	0.4883 (0.2912)	0.4478 (0.2692)	0.4967 (0.2659)	0.4826 (0.2572)	0.5201 (0.2964)
	AUC	0.7110 (0.1845)	0.7221 (0.1961)	0.6967 (0.1728)	0.7183 (0.1565)	0.7397 (0.1980)
	nDCG	0.7161 (0.3107)	0.6615 (0.2709)	0.7686 (0.3001)	0.7364 (0.276)	0.7401 (0.2597)
Supervised	RA	0.7717 (0.1960)	0.7175 (0.2098)	0.7988 (0.1906)	0.7741 (0.1945)	0.7899 (0.1789)
	MAP	0.5659 (0.3384)	0.5966 (0.3219)	0.6854 (0.3001)	0.6324 (0.3064)	0.6963 (0.3103)
	AUC	0.8242 (0.1857)	0.8003 (0.2225)	0.8585 (0.1802)	0.8306 (0.1954)	0.8813 (0.1591)
	nDCG	0.6550 (0.4195)	0.6928 (0.3984)	0.7717 (0.3635)	0.7324 (0.3724)	0.7518 (0.3669)
		Set II				
		CB	FP	MCS	LPCS	SW
Unsupervised	RA	0.7216 (0.1911)	0.7212 (0.1546)	0.8070 (0.1736)	0.7515 (0.1647)	0.7856 (0.1591)
	MAP	0.9156 (0.1662)	0.7156 (0.2478)	0.9094 (0.1514)	0.8303 (0.1768)	0.9167 (0.1634)
	AUC	0.9415 (0.1209)	0.8714 (0.1447)	0.9622 (0.0847)	0.8937 (0.1165)	0.9523 (0.1146)
	nDCG	0.9599 (0.0708)	0.8189 (0.2003)	0.9330 (0.0948)	0.8986 (0.1146)	0.9589 (0.0768)
Supervised	RA	0.9997 (0.0011)	0.931 (0.1096)	0.9944 (0.0253)	0.9883 (0.0583)	0.9985 (0.0076)
	MAP	0.9995 (0.0037)	0.905 (0.2397)	0.9980 (0.0265)	0.9663 (0.1172)	0.9980 (0.0393)
	AUC	0.9999 (0.0005)	0.9844 (0.0502)	0.9997 (0.0036)	0.9926 (0.0354)	0.9998 (0.0042)
	nDCG	0.9983 (0.0185)	0.8887 (0.2243)	0.9744 (0.0773)	0.9467 (0.1183)	0.9968 (0.0344)

four cross-validation folds of equal size. In the unsupervised case each subset was used individually to allow for a comparison with the supervised model. Of such a subset, each enzyme was used as a query to rank the remaining enzymes, as described in Section 2.3. The performance for each of these rankings was averaged to obtain the global performance over the folds.

In the supervised setting, each fold was withheld as a test set, while the other three parts of the data set were used for training and model selection. This process was repeated for each part, so that every instance was used for training and testing (thus, four-fold outer cross-validation). Neither the query nor the database enzymes are thus used for building a model, which allows us to demonstrate that our methods can generalize to new enzymes. In addition, a 10-fold inner cross-validation loop was implemented for estimating the optimal regularization parameter λ , as recommended in [75]. The value of this hyperparameter, which controls model complexity, was selected from a grid containing all powers of 10 from 10^{-4} to 10^5 . The final model was trained using the whole training set and the median of the best hyperparameter values over the ten folds. We used the implementation RLScore³ in Python to train the models.

3 Results and discussion

3.1 Differences between cavity-based similarities and data sets

Table 2 gives a global summary of the results obtained for the unsupervised and the supervised ranking approach for both data sets. One can note a sizeable difference between the performances for the different cavity-based similarities, data sets and the performance measures used. Despite this variation, it is clear that data set I is considerably harder than data set II. This can easily be explained by the fact that data set II only contains enzymes with a certain resolution of the active site. Furthermore, for set II the active site is determined by an expert, while for set I the active site is resolved by heuristically choosing the largest cavity. It is likely that some mistakes are made in this annotation process. Consequently, inferring functional similarity of data set I will be harder.

The cavity-based similarity measure based on fingerprints usually results in the worst performance, except for the ranking error in the unsupervised setting. It seems that the performance of the FP does not improve as much in the supervised approach, compared to other cavity-based similarity measures. This is likely because the fingerprints cause a high loss of information, since even functionally dissimilar enzyme cavities can be considered similar according to this metric.

Comparing the two graph-based similarities (MCS and CB), we see some differences between the data sets. Though both perform relatively well, MCS performs better for data set I, while CB is the clear champion of data

³See <http://staff.cs.utu.fi/~aatapa/software/RLScore/> for this software.

set II. The good performance of CavBase for data set II can be explained easily. CavBase computes the 100 largest common subgraphs, which could be used to construct a cavity-based similarity measure. However, a graph representation leads to a loss of information, since the coordinates of pseudocenters cannot be restored. Moreover, since the size of the (maximum) common subgraph is an integer that usually lies in the range of 4 to 12 nodes, there is a loss of resolution by mapping many different pairs of cavities to the same similarity score. In theory, MCS suffers from these drawbacks. Even though the resolution problem is to a certain extent solved if the size of the maximum common subgraph is divided by the size of the larger binding site, the graph representation could still lead to a slight loss of information.

On the other hand, the LPCS measure uses geometric information, hence, no loss of information is introduced by transforming the pseudocenter representation into a graph representation. Moreover, this transformation does not cause the resolution problem. Yet, the measure is computed via solving a multimodal optimization problem, so it is possible to get stuck in a local optimum, resulting in a similarity score that is too low. Similar to MCS, LPCS seems to perform relatively better for data set I compared to data set II, probably because the local optimum becomes less of an issue in the former case. This can be explained by the fact that data set I contains larger cavities, on average, hence making it harder to find the global optimum.

Finally, we consider the measure based on sequence alignment. For data set I, the SW similarity measure competes with the MCS as one of the best measures, depending on the performance measure. For data set II in the supervised case, it is only outperformed by the CB measure. It is clear that the SW measure is a powerful method for comparing cavities, as it is also limited by a bad resolution of the cleft.. Like MCS, SW seeks to quantify the largest similar region, here as a local alignment. As this contains, information about the common residues of the cavity, this is a simple, though powerful measure.

3.2 The benefits of supervised ranking

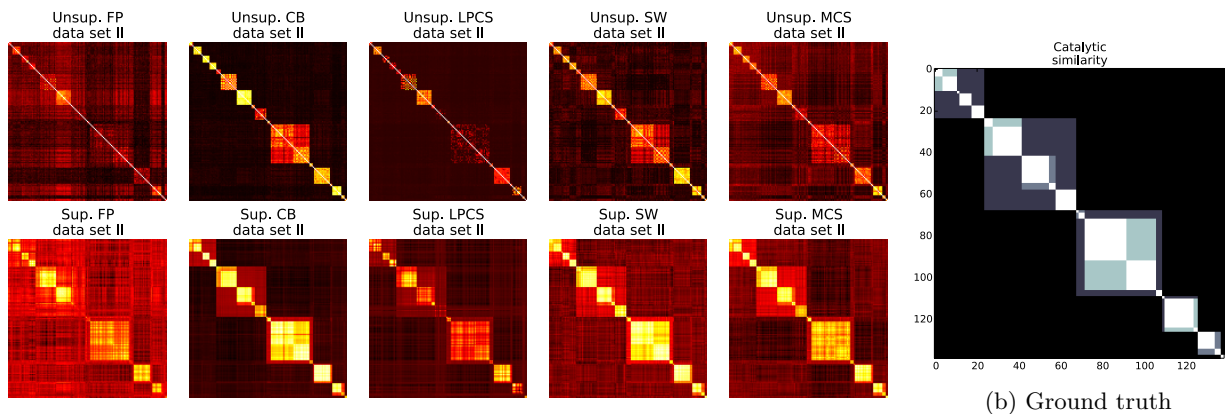
The ranking of both data set I and data set II showed a considerable improvement in the supervised approach. Three important reasons can be put forward to explain the improvement in performance. In this section we will illustrate this using data set II, as this one showed the most clear effects of learning. First of all, the traditional benefit of supervised learning plays an important role. One can expect that supervised ranking methods outperform unsupervised methods, because they take annotations into account during the training phase to guide the model towards retrieval of enzymes with a similar EC number. Conversely, unsupervised methods solely rely on a meaningful similarity measure between enzymes, while ignoring EC numbers.

Second, we also advocate that supervised ranking methods have the ability to preserve the hierarchical structure of EC numbers in their predicted rankings. Figure 2 supports this claim. It summarizes the values used for ranking one fold of the test set obtained by the different models. A higher value (indicated by a lighter color) in a row means that this enzyme is considered to have a higher catalytic similarity w.r.t. the query enzyme. So, for unsupervised ranking it visualizes $K^\phi(v, v')$, for supervised ranking the values $h(v, v')$ are shown. Each row of the heatmap corresponds to one query. For the supervised models one notices a much better correspondence with the ground truth. Furthermore, the different levels of catalytic similarity can be better distinguished. In addition, an example of the distributions of the predicted values within one query is visualized in Figure 3 by means of box plots. The different populations within a plot correspond to the different levels of the catalytic similarity w.r.t. the query enzyme. This illustrates again that supervised models can make a better discrimination between enzymes that are functionally more similar and those that are dissimilar. For this example query, no quartiles are overlapping in any supervised model, unlike the unsupervised approach, which only detects a good ranking for exact matches (i.e., enzymes having an EC number identical to the query).

A third reason for improvement by the supervised ranking method can be found in the exploitation of dependencies between different catalytic similarity values. Roughly speaking, if one is interested in the catalytic similarity between enzymes v and v' , one can try to compute this catalytic similarity in a direct way based on mutual relationships in cavities, or derive it in an indirect way from the cavity-based similarity with a third enzyme v'' . This division into a direct and an indirect approach shows a certain correspondence to similar discussions in the context of inferring protein-protein interaction and signal transduction networks – see e.g. [76, 70, 77]. Unsupervised ranking boils in a certain sense down to a direct approach, while supervised ranking should be interpreted as indirect. Especially when the similarity matrix contains noisy values, one can expect that the indirect approach allows for detecting *back bone* entries and correcting the noisy ones.

3.3 Differences between performance measures

Table 2 indicates that the different performances are to some degree influenced by the similarity measure and data set used. This is especially clear for the supervised ranking approach. One can observe a very clear distinction



(a) Unsupervised and supervised measures used for ranking

Figure 2: (a) Heatmaps of the values used for ranking the data set II for one fold in the testing phase. Each row of the heatmap corresponds to one query. The four figures on top visualize the cavity-based similarities $K^\phi(v, v')$ that are used to construct an unsupervised ranking. The four figures at the bottom visualize the model output $h(v, v')$, which is used to derive a supervised ranking. (b) The ground truth catalytic similarity that had to be learned.

between the ranking accuracy and area under the ROC curve, which treat every position as equally important, and the other two measures, which emphasize the top of the ranking. This should not come as a surprise, as an approximation of the ranking error is optimized in our algorithms. The AUC is very related to the RA, as they coincide for bipartite rankings. In the latter we only make a distinction between 'relevant' enzymes, which have three or more EC numbers in common, and enzymes which do not. Since the RA uses a finer fragmentation of functional similarity, this is a more severe performance measure compared to the AUC. For data set II, both AUC and RA are both very close to the theoretical optimum for nearly all cavity-based similarity measures in the supervised case.

Figure 4 shows the ROC curves that are obtained by applying the cut-off threshold for data set I, which defines a database enzyme a hit if at least the first three digits of the EC number are correct. In contrast to the scalar performance measures of Table 2, the ROC curve gives information about the quality of the ranking at all positions. It is immediately clear that supervised ranking outcompetes unsupervised ranking, as the former's curves are closer to the upper left corner. Typically for these curves, there is a part in the beginning where the line has a very high slope, showing that a certain fraction of relevant objects can be detected with very high sensitivity and specificity. This fraction that can be detected nearly without mistakes increases after the supervised learning step, as indicated by the higher 'offset' of these curves (most clear for SW, LPCS and FP in the ROC curve). The next section of the curve is usually a part with a smaller average slope, indicating that at some point it becomes harder to detect signal. For the unsupervised curves, this is nearly a straight line, which means that from that point the detection of catalytically similar enzymes is essentially random. The supervised curves still have a concave shape at their second part, which shows that relevant enzymes can still be detected in that piece.

From Table 2 it is also clear that supervised ranking usually scores worse for MAP and nDCG compared to RA and AUC. For the nDCG the performance sometimes even decreases after learning a model! This can easily be explained by the fact that our model optimizes the quality of the complete ranking, in contrast to only the top which is assessed by MAP and nDCG. Note that the top functionally similar enzymes (i.e. the same EC number) can likely be detected based on the cavity-based similarity alone. Hence, training a model might not be required to perform good in this section. One can see this learning effect nicely in Figure 4 for the CB similarity measure. The quality of the supervised ranking of the top for this measure is worse than for all other measures (indicated by the low nDCG). The overall ranking (indicated by the AUC) is quite good in comparison, as the lower part compensates for the bad ranking at the top. Depending on the application, the top or the general ranking might be more of interest.

4 Related work

Since the comparison of enzymes has become an important task in functional bioinformatics, a vast number of similarity measures for proteins has been proposed so far. As mentioned in the introduction, a reliable method

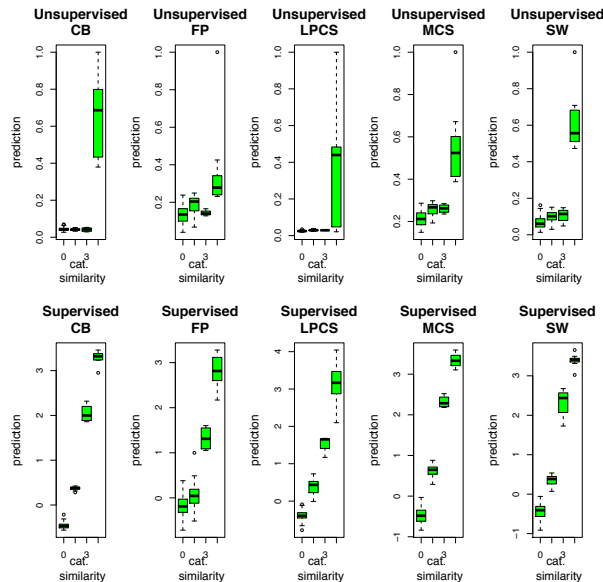


Figure 3: Box-and-whisker plots of the values used for ranking data set II for one randomly chosen query as an example. The different populations on the x-axis denote the groups that are formed by subdividing the database enzymes according to the number of EC number digits they share with the query. Given a query v and a database enzyme v' , the y-axis shows the distribution of the values $h(v, v')$ and $K^\phi(v, v')$ for the supervised and the unsupervised approach, respectively. For nearly all cavity-based similarity measures, one can observe a much better separation of the groups for the supervised approach.

will focus on the geometry and the physico-chemical properties of certain regions of an enzyme. However, methods that are based on the sequence or the fold usually exhibit a much lower complexity and they can also lead to good results, especially when the sequence identity is above a certain threshold. *ProFunc* [78] is in this regard a very interesting tool, in which a bulk of different methods is applied, such as sequence alignment, motif and template search, and also a comparison of active sites. The biological function of enzymes is derived from the closest match in different databases such as PDB, UniProt and PROCAT, and finally returned by the program. Despite being very powerful, this approach becomes nevertheless very inefficient, with runtimes up to several hours for a single protein. Since we considered a sizeable data set for which nearly 3,000,000 pairwise similarity scores had to be computed, it became impossible to compare our results to ProFunc.

In addition to focusing on individual enzymes, one can also take protein-protein interactions into account for inferring the function. Proteins that are close to each other in a protein-protein interaction network are expected to have similar functions, so one can try to infer the function of an unannotated protein by looking at its neighbors [79]. Similarly, one can also solve optimization problems over the global network, such as maximizing the number of edges that connect proteins sharing the same function [80, 81]. Other approaches make use of probabilistic graphical models such as Markov random fields [82, 83]. Conceptually, these methods might also enrich the predictions obtained by an unsupervised search-based approach, but they usually do not consider cavity and binding site information to predict the function of proteins.

5 Conclusion

In this paper we have recast the EC annotation problem as a conditional ranking problem. We have shown that retrieval of enzymes w.r.t. functionality can be substantially improved by applying a supervised ranking method that takes advantage of ground truth EC numbers during the training phase. In contrast, more traditional methods rely heavily on a notion of similarity to search for functionally related enzymes in online databases. Such methods lead to an unsupervised approach in which annotations are not taken into account.

We focused specifically on cavity-based similarity measures, because their benefits compared to sequence-based and structure-based approaches have been demonstrated in previous work, although our method can work with any meaningful similarity measure defined on enzymes. In the experiments we could demonstrate a considerable improvement of the quality of the overall ranking. The results were influenced by the type of data used and the way

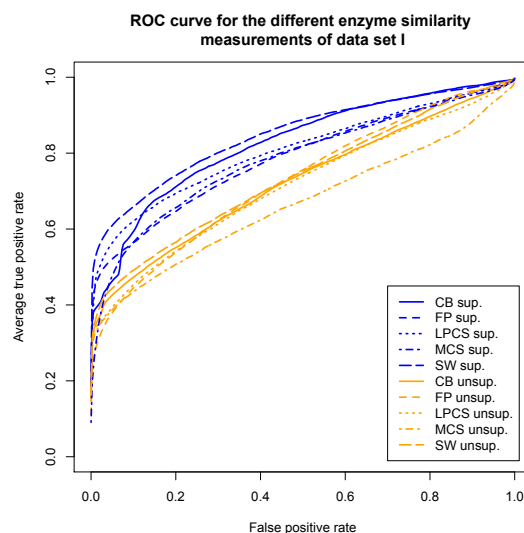


Figure 4: Receiver operating characteristic curves for the unsupervised and supervised ranking methods for data set I. An enzyme is considered functionally similar to the query if the first three digits of the EC number are identical to those of the query.

the ranking was evaluated, indicating that the most optimal method is highly dependent of the specific problem setting. Nevertheless, our supervised ranking algorithm outperformed the unsupervised ranking algorithm for all cavity-based similarities for most performance measures considered. While the unsupervised approach succeeded quite well in returning exact matches to a query, the hierarchical structure of EC numbers was better preserved in the rankings predicted by the supervised approach. As such, supervised ranking can be interpreted as a powerful alternative for retrieval methods that are more traditionally used in bioinformatics.

Acknowledgements

MS, BDB and WW acknowledge the support of Ghent University (MRP Bioinformatics: from nucleotides to networks). T.F., E.H., S.G. and G.K. gratefully acknowledge financial support by the German Research Foundation (DFG) and the LOEWE Research Center for Synthetic Microbiology, Marburg. T.P. and A.A. are both supported for this work by the Academy of Finland (grant 134020 and 128061, respectively).

References

- [1] I. Friedberg, “Automated protein function prediction—the genomic challenge,” *Briefings in Bioinformatics*, vol. 7, pp. 225–42, Sept. 2006.
- [2] D. Dunaway-Mariano, “Enzyme function discovery,” *Structure*, vol. 16, no. 11, pp. 1599–1600, 2008.
- [3] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, “Basic Local Alignment Search Tool,” *Journal of Molecular Biology*, vol. 215, no. 3, pp. 403–410, 1990.
- [4] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, “Gapped Blast and PsiBlast: a new generation of protein database search programs,” *Nucleic Acid Research*, vol. 25, no. 17, pp. 3389–3402, 1997.
- [5] C. Leslie, E. Eskin, and W. S. Noble, “The spectrum kernel: a string kernel for SVM protein classification,” in *Pacific Symposium on Biocomputing*, pp. 566–575, Jan. 2002.
- [6] C. S. Leslie, E. Eskin, A. Cohen, J. Weston, and W. S. Noble, “Mismatch string kernels for discriminative protein classification,” *Bioinformatics*, vol. 20, pp. 467–76, Mar. 2004.

- [7] R. Powers, J. C. Copeland, K. Germer, K. A. Mercier, V. Ramanathan, and P. Revesz, "Comparison of protein active site structures for functional annotation of proteins and drug design," *Proteins: Structure, Function, and Bioinformatics*, vol. 65, pp. 124–135, 2006.
- [8] B. Rost, "Enzyme function less conserved than anticipated," *Journal of Molecular Biology*, vol. 318, no. 2, pp. 595–608, 2002.
- [9] A. J. Chalk, C. L. Worth, J. P. Overington, and A. W. E. Chan, "PDBLIG: Classification of small molecular protein binding in the protein data bank," *Journal of Medical Chemistry*, vol. 47, no. 15, pp. 3807–3816, 2004.
- [10] K. Kinoshita, Y. Murakami, and H. Nakamura, "eF-seek: prediction of the functional sites of proteins by searching for similar electrostatic potential and molecular surface shape," *Nucleic Acid Research*, vol. 35, no. suppl 2, pp. W398–W402, 2007.
- [11] J. Thornton, A. Todd, D. Milburn, N. Borkakoti, and C. Orengo, "From structure to function: Approaches and limitations," *Nature Structural Biology*, vol. 7 Suppl, pp. 991–994, 2000.
- [12] M. Kotera, Y. Okuno, M. Hattori, S. Goto, and M. Kanehisa, "Computational assignment of the EC numbers for genomic-scale analysis of enzymatic reactions," *Journal of the American Chemical Society*, vol. 126, pp. 16487–98, Dec. 2004.
- [13] V. Egelhofer, I. Schomburg, and D. Schomburg, "Automatic assignment of EC numbers," *PLoS Computational Biology*, vol. 6, pp. 1000661+, 2010.
- [14] M. Shatsky, R. Niussinov, and H. J. Wolfson, "A method for simultaneous alignment of multiple protein structures," *Proteins: Structure, Function, and Bioinformatics*, vol. 56, pp. 143–156, 2004.
- [15] M. Shatsky, *The Common Point Set Problem with Applications to Protein Structure Analysis*. PhD thesis, Tel Aviv University, Tel Aviv, Israel, 2006.
- [16] A. Harrison, F. Pearl, I. Sillitoe, T. Slidel, R. Mott, J. Thornton, and C. Orengo, "Recognizing the fold of a protein structure," *Bioinformatics*, vol. 19, no. 14, pp. 1748–1759, 2003.
- [17] A. C. Martin, C. A. Orengo, E. G. Hutchinson, S. Jones, M. Karmirantzou, R. A. Laskowski, J. B. Mitchell, C. Taroni, and J. M. Thornton, "Protein folds and functions," *Structure*, vol. 6, pp. 875–84, July 1998.
- [18] P. C. Babbitt and J. Gerlt, "Understanding enzyme superfamilies," *Journal of Biological Chemistry*, vol. 272, pp. 30591–30594, Dec. 1997.
- [19] J. A. Gerlt and P. C. Babbitt, "Divergent evolution of enzymatic function: mechanistically diverse superfamilies and functionally," *Annual review of biochemistry*, vol. 70, pp. 209–246, 2001.
- [20] M. Osadchy and R. Kolodny, "Maps of protein structure space reveal a fundamental relationship between protein structure and function," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 30, pp. 12301–12306, 2011.
- [21] D. M. Kristensen, R. M. Ward, A. M. Lisewski, S. Erdin, B. Y. Chen, V. Y. Fofanov, M. Kimmel, L. E. Kaviraki, and O. Lichtarge, "Prediction of enzyme function based on 3D templates of evolutionarily important amino acids," *BMC Bioinformatics*, vol. 9, Jan. 2008.
- [22] S. Erdin, A. M. Lisewski, and O. Lichtarge, "Protein function prediction: towards integration of similarity metrics," *Current Opinion in Structural Biology*, vol. 21, pp. 180–8, Apr. 2011.
- [23] R. A. Laskowski, N. M. Luscombe, M. B. Swindells, and J. M. Thornton, "Protein clefts in molecular recognition and function," *Protein Science*, vol. 5, no. 12, pp. 2438–2452, 1996.
- [24] S. Pérot, O. Sperandio, M. A. Miteva, A.-C. Camproux, and B. O. Villoutreix, "Druggable pockets and binding site centric chemical space: a paradigm shift in drug discovery," *Drug Discovery Today*, vol. 15, no. 15-16, pp. 656–667, 2010.
- [25] C. Andersson, B. Y. Chen, and A. Linusson, "Mapping of ligand-binding cavities in proteins," *Proteins*, vol. 78, no. 6, pp. 1408–1422, 2010.

- [26] M. Weisel, J. M. Kriegel, and G. Schneider, “Architectural repertoire of ligand-binding pockets on protein surfaces,” *Chembiochem*, vol. 11, no. 4, pp. 556–563, 2010.
- [27] A. Weber, A. Casini, A. Heine, D. Kuhn, C. T. Supuran, A. Scozzafava, and G. Klebe, “Unexpected nanomolar inhibition of carbonic anhydrase by COX-2-selective celecoxib: new pharmacological opportunities due to related binding site recognition,” *Journal of Medical Chemistry*, vol. 47, no. 3, pp. 550–557, 2004.
- [28] J. Huan, D. Bandyopadhyay, W. Wang, J. Snoeyink, J. Prins, and A. Tropsha, “Comparing graph representations of protein structure for mining family-specific residue-based packing motifs,” *Journal of computational biology : a journal of computational molecular cell biology*, vol. 12, no. 6, pp. 657–71, 1999.
- [29] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. V. N. Vishwanathan, A. J. Smola, and H.-P. Kriegel, “Protein function prediction via graph kernels,” *Bioinformatics*, vol. 21 Suppl 1, pp. i47–56, June 2005.
- [30] T. Gärtner, *Kernels for Structured Data*. Singapore: World Scientific, 2008.
- [31] N. Shervashidze, “Efficient graphlet kernels for large graph comparison,” in *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 5, pp. 488–495, 2009.
- [32] V. Vacic, L. I. ans S. Lonardi, and P. Radivojac, “Graphlet kernels for prediction of functional residues in protein structures,” *Journal of Computational Biology*, vol. 17, no. 1, pp. 55–72, 2010.
- [33] M. Shatsky, A. Shulman-Peleg, R. Nussinov, and H. J. Wolfson, “The multiple common point set problem and its application to molecule binding pattern detection,” *Journal of Computational Biology*, vol. 13, no. 2, pp. 407–428, 2006.
- [34] T. Fober, S. Glinca, G. Klebe, and E. Hüllermeier, “Superposition and alignment of labeled point clouds,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, pp. 1653–1666, November/December 2011.
- [35] N. Weill and D. Rognan, “Alignment-free ultra-high-throughput comparison of druggable protein-ligand binding sites,” *Journal of Chemical Information and Modeling*, vol. 50, no. 1, pp. 123–135, 2010.
- [36] T. Fober, M. Mernberger, G. Klebe, and E. Hüllermeier, “Fingerprint kernels for protein structure comparison,” *Molecular Informatics*, vol. 31, no. 6-7, pp. 443–452, 2012.
- [37] E. Hüllermeier and J. Fürnkranz, *Preference Learning*. Springer, 2010.
- [38] F. Rathke, K. Hansen, U. Brefeld, and K.-R. Müller, “StructRank: a new approach for ligand-based virtual screening,” *Journal of Chemical Information and Modeling*, vol. 51, pp. 83–92, Dec. 2010.
- [39] S. Agarwal, D. Dugar, and S. Sengupta, “Ranking chemical structures for drug discovery: a new machine learning approach,” *Journal of Chemical Information and Modeling*, vol. 50, pp. 716–31, May 2010.
- [40] J. Weston, A. Eliseeff, D. Zhou, C. Leslie, and W. S. Noble, “Protein ranking: from local to global structure in the protein similarity network,” *Proceedings of the National Academy of Science*, vol. 101, pp. 6559–6563, 2004.
- [41] R. Kuang, J. Weston, W. S. Noble, and C. Leslie, “Motif-based protein ranking by network propagation,” *Bioinformatics*, vol. 21, pp. 3711–8, Oct. 2005.
- [42] T. Pahikkala, E. Tsivtsivadze, A. Airola, J. Järvinen, and J. Boberg, “An efficient algorithm for learning to rank from preference graphs,” *Machine Learning*, vol. 75, no. 1, pp. 129–165, 2009.
- [43] T. Joachims, “A support vector method for multivariate performance measures,” in *Proceedings of the International Conference on Machine Learning, Bonn, Germany*, pp. 377–384, 2005.
- [44] P. D. Dobson and A. J. Doig, “Predicting enzyme class from protein structure without alignments,” *Journal of Molecular Biology*, vol. 345, pp. 187–99, Jan. 2005.
- [45] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor, “Kernel-based learning of hierarchical multilabel classification models,” *Journal of Machine Learning Research*, vol. 7, pp. 1601–1626, 2006.
- [46] A. Sokolov and A. Ben-Hur, “A structured-outputs method for prediction of protein function,” in *Proceedings of the 3rd International Workshop on Machine Learning in Systems Biology*, 2008.

- [47] A. K. Arakaki, Y. Huang, and J. Skolnick, “EFICAz2: enzyme function inference by a combined approach enhanced by machine learning,” *BMC Bioinformatics*, vol. 10, p. 107, Jan. 2009.
- [48] M. Hendlich, A. Bergner, J. Günther, and G. Klebe, “Relibase: design and development of a database for comprehensive analysis of proteinligand interactions,” *Journal of Molecular Biology*, vol. 326, pp. 607–620, Feb. 2003.
- [49] S. Schmitt, D. Kuhn, and G. Klebe, “A new method to detect related function among proteins independent of sequence and fold homology,” *Journal of Molecular Biology*, vol. 323, no. 2, pp. 387–406, 2002.
- [50] H. Bunke and K. Shearer, “A graph distance metric based on the maximal common subgraph,” *Pattern Recognition Letters*, vol. 19, no. 3-4, pp. 255–259, 1998.
- [51] A. Sanfeliu and K. Fu, “A distance measure between attributed relational graphs for pattern recognition,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 13, no. 3, pp. 353–362, 1983.
- [52] N. Weskamp, E. Hüllermeier, D. Kuhn, and G. Klebe, “Multiple graph alignment for the structural analysis of protein active sites,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 2, pp. 310–20, 2007.
- [53] M. Neuhaus and H. Bunke, *Bridging the Gap between Graph Edit Distance and Kernel Machines*. New Jersey: World Scientific, 2007.
- [54] K. Borgwardt, C. Ong, S. Schonauer, S. Vishwanathan, A. Smola, and H. Kriegel, “Protein function prediction via graph kernels,” *Bioinformatics*, vol. 21, no. 1, pp. i47–i56, 2005.
- [55] H. Alt and L. J. Guibas, “Discrete geometric shapes: Matching, interpolation, and approximation: A survey,” tech. rep., *Handbook of Computational Geometry*, 1996.
- [56] P. Mahé, N. Ueda, T. Akutsu, J.-L. Perret, and J.-P. Vert, “Graph kernels for molecular structure-activity relationship analysis with support vector machines,” *Journal of chemical information and modeling*, vol. 45, no. 4, pp. 939–51, 2005.
- [57] M. M. Deza and E. Deza, *Encyclopedia of Distances*. Heidelberg, Germany: Springer, 2009.
- [58] H.-G. Beyer and H.-P. Schwefel, “Evolution strategies: A comprehensive introduction,” *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.
- [59] B. Hoffmann, M. Zaslavskiy, J.-P. Vert, and V. Stoven, “A new protein binding pocket similarity measure based on comparison of clouds of atoms in 3D: application to ligand prediction,” *BMC Bioinformatics*, vol. 11, Jan. 2010.
- [60] T. Fober, M. Mernberger, G. Klebe, and E. Hüllermeier, “Evolutionary construction of multiple graph alignments for the structural analysis of biomolecules,” *Bioinformatics*, vol. 25, pp. 2110–7, Aug. 2009.
- [61] W. Kabsch, “A solution of the best rotation to relate two sets of vectors,” *Acta Crystallographica*, vol. 32, pp. 922–923, 1976.
- [62] T. Fober, M. Mernberger, R. Moritz, and E. Hüllermeier, “Graph-kernels for the comparative analysis of protein active sites,” in *German Conference on Bioinformatics*, (Halle (Saale), Germany), pp. 21 – 31, 2009.
- [63] T. F. Smith and M. S. Waterman, “Identification of common molecular subsequences,” *Journal of Molecular Biology*, vol. 147, pp. 195–197, 1981.
- [64] T. Pahikkala, W. Waegeman, E. Tsivtsivadze, T. Salakoski, and B. De Baets, “Learning intransitive reciprocal relations with kernel methods,” *European Journal of Operational Research*, vol. 206, pp. 676–685, Nov. 2010.
- [65] T. Pahikkala, A. Airola, M. Stock, B. De Baets, and W. Waegeman, “Efficient regularized least-squares algorithms for conditional ranking on relational data,” *Machine Learning*, vol. 93, no. 2-3, pp. 321–356, 2013.
- [66] B. Schölkopf and A. Smola, *Learning with Kernels, Support Vector Machines, Regularisation, Optimization and Beyond*. The MIT Press, 2002.
- [67] W. Waegeman, T. Pahikkala, A. Airola, T. Salakoski, M. Stock, and B. De Baets, “A kernel-based framework for learning graded relations from data,” *IEEE Transactions on Fuzzy Systems*, vol. 20, pp. 1090–1101, 2012.

- [68] A. Ben-Hur and W. S. Noble, “Kernel methods for predicting protein-protein interactions,” *Bioinformatics*, vol. 21, pp. i38–46, June 2005.
- [69] H. Kashima, S. Oyama, Y. Yamanishi, and K. Tsuda, “On pairwise kernels: An efficient alternative and generalization analysis,” in *PAKDD* (T. Theeramunkong, B. Kijssirikul, N. Cercone, and T. B. Ho, eds.), vol. 5476 of *Lecture Notes in Computer Science*, pp. 1030–1037, Springer, 2009.
- [70] J.-P. Vert, J. Qiu, and W. S. Noble, “A new pairwise kernel for biological network inference with support vector machines,” *BMC Bioinformatics*, vol. 8, Jan. 2007.
- [71] R. Herbrich, T. Graepel, and K. Obermayer, “Large margin rank boundaries for ordinal regression,” in *Advances in Large Margin Classifiers* (A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, eds.), pp. 115–132, MIT Press, 2000.
- [72] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. MIT Press, 2006.
- [73] W. Waegeman, B. De Baets, and L. Boullart, “ROC analysis in ordinal regression learning,” *Pattern Recognition Letters*, vol. 29, pp. 1–9, 2008.
- [74] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of IR techniques,” *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.
- [75] S. Varma and R. Simon, “Bias in error estimation when using cross-validation for model selection,” *BMC Bioinformatics*, vol. 7, Jan. 2006.
- [76] R. Albert, B. DasGupta, R. Dondi, S. Kachalo, E. Sontag, A. Zelikovsky, and K. Westbrook vol. 14, pp. 927–949, 2007.
- [77] P. Geurts, N. Touleimat, M. Dutreix, and F. D’Alché-Buc, “Inferring biological networks with output kernel trees,” *BMC Bioinformatics*, vol. 8, no. 2, p. S4, 2007.
- [78] R. Laskowski, J. Watson, and J. Thornton, “Profunc: a server for predicting protein function from 3d structure,” *NAR*, vol. 33, pp. W89–W93, 2005.
- [79] B. Schwikowski, P. Uetz, and S. Fields, “A network of protein-protein interactions in yeast,” *Nature Biotechnology*, vol. 18, pp. 3548–3557, 2000.
- [80] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani, “Global protein function prediction from protein-protein interaction networks,” *Bioinformatics*, vol. 21, pp. ii59–ii65, 2003.
- [81] U. Karaoz, T. Murali, S. Letovsky, Y. Zheng, C. Ding, C. Cantor, and S. Kasif, “Whole-genome annotation by using evidence integration in functional-linkage networks,” *Proceedings of the National Academy of Sciences*, vol. 101, pp. 2888–2893, 2004.
- [82] M. Deng, K. Zhang, S. Mehta, T. Chen, and F. Sun, “Prediction of protein function using protein-protein interaction data,” *Journal of Computational Biology*, vol. 10, pp. 947–960, 2003.
- [83] S. Letovsky and S. Kasif, “Predicting protein function from protein/protein interaction data: a probabilistic approach,” *Bioinformatics*, vol. 19, pp. i197–i204, 2003.